Title	ワンタイムパスワードによるパターン認証を用いたパスワードマネージャの提案
Author(s)	島田，憲吾
Citation	
Date of issue	2018-03
URL	http://hdl.handle.net/10173/1970
Rights	
Text version	author

平成 29 年度

学士学位論文

ワンタイムパスワードによるパターン認証を用いたパスワードマネージャの提案

Suggestion of the password manager using the pattern certification by the one-time password

1180338　　島田　憲吾

指導教員　　清水　明宏

2018 年 2 月 28 日

高知工科大学 情報学群

要 旨

ワンタイムパスワードによるパターン認証を用いたパスワードマネージャの提案

島田　憲吾

近年, インターネットの普及に伴い, Web サービスの利用人口は増加しており, 複数の Web サービスで同じ ID・パスワードを使いまわしているユーザが増加傾向にある. それを解決するためにマスターパスワード 1 つで管理するパスワードマネージャが存在するが, マスターパスワードの保護が重要になる. 既存方式では, パターン認証とユーザ証明書を用いて 2 要素認証を行うパスワードマネージャが提案されている. しかし, 既存方式では生成される認証情報が静的であるため安全性に懸念があること, Web アプリケーションとして提案されているが複数の端末で利用できない 2 つの問題がある.

本研究では, 認証毎に異なる認証情報を用いて認証を行うことで安全性を向上し, 複数端末化を実現することで利便性を向上する.

キーワード　パスワードマネージャ, パスワード, Web サービス, パターン認証, ワンタイムパスワード, QR コード

Abstract

Suggestion of the password manager using the pattern certification by the one-time password

Kengo Shimada

In late years, with the spread of Internet, the use population of the Web Service increases, and there is a user recycling the same ID, password by plural Web Service in an increase tendency. A password manager to manage with one master password to solve it exists, but the protection of the master password becomes important. By the existing method, two elements password manager authenticating is suggested to using the pattern certification and a user certificate. However, of two that is suggested to safety as a thing, the Web application that there is concern because generated certification information is static by the existing method, but are not available at plural terminals have a problem. In this study, I improve safety by authenticating it using different certification information every certification and improve convenience because a plural number terminal realizes making it it.

key words Password manager, Password, Web Service, Pattern lock, One-time password, QR code

目次

図目次

表目次

第 1 章

はじめに

近年, インターネットの普及に伴い, Web サービスの利用人口が増加している. その中でも SNS(Social Networking Service) や動画視聴サイトなど様々な種類の Web サービスが利用されている.

図 1.1　Web サービス, SNS の利用人口推移

複数の Web サービスを利用するに当たって, 各 Web サービスの ID・パスワードを管理することが困難になってしまう. それによって, 複数の Web サービスで同じ ID・パスワードを使いまわしているという問題が浮上してきた. 同じ ID・パスワードの使いまわしは, パスワードリスト攻撃による個人情報の漏洩の危険性があるため, 推奨されていない。　その問題を解決するためにパスワードマネージャというシステムが存在する。パスワードマネージャとは, 複数の Web サービスの ID・パスワードをマスターパスワード 1 つで一括管理するソフトウェアである. Web サービスを利用する際, マスターパスワードを用いてパスワードマネージャに問い合わせ, Web サービスの ID・パスワードを取り出す. マスターパスワー

ド 1 つで管理するため, マスターパスワードの保護が重要になる.　しかし, キーロガー等による ID・ パスワードの不正窃取により, マスターパスワードが漏洩してしまった場合, 複数の Web サービスの個人情報が漏洩する可能性があり, 被害が大きくなってしまう. 既存方式では, パターン認証とユーザ証明書を用いることで, キーロガーによる不正窃取に対して耐性を兼ね備えたパスワードマネージャが提案されているが, 生成された認証情報は静的な認証情報となっており, マスターパスワードが推測されてしまう危険性がある. SaaS 型のパスワードマネージャでは, 高度なセキュリティが求められるため, 静的なパスワードでは安全性に懸念が残る. それだけでなく, 既存方式は複数の端末で利用することができない. 複数の端末を利用するユーザは増加傾向にあるが, 既存方式のように 1 つの端末でしか利用できないパスワードマネージャでは利便性の低下を招いてしまう恐れがある.　本研究では, 既存方式と同等の攻撃に対して耐性を持ちつつ, 認証情報を認証毎にワンタイムに変更することでより安全性を向上したパスワードマネージャを提案する. また, QR コードを用いて端末間で, 安全に認証に必要な情報を共有することで, 複数端末の利用を実現し利便性を向上する.

第 2 章

既存方式

2.1　パスワードマネージャ

パスワードマネージャとは, 複数の Web サービスの ID・パスワードをサーバやローカルストレージに保存し, マスターパスワードというパスワードで一括管理するソフトウェアである. 図??にソフトウェアの概略図を示す.

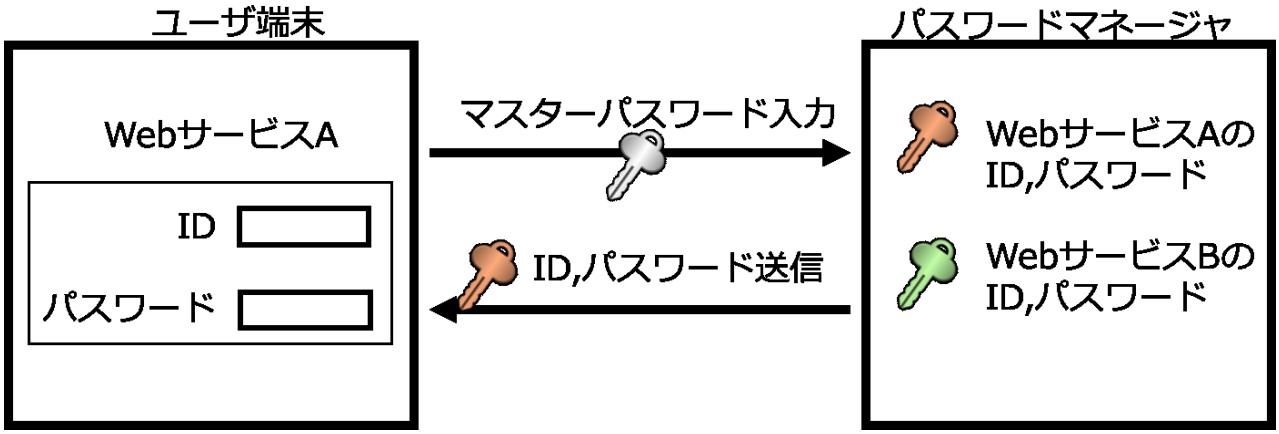

図 2.1　パスワードマネージャの概略図

パスワードマネージャには事前に, 複数の Web サービスの ID・パスワードを保存しておく. Web サービス A を利用する際に, マスターパスワードでパスワードマネージャに問い合わせる. Web サービス A の ID・パスワードを取り出すことで, サービスを利用できるようになる. マスターパスワード 1 つで複数の Web サービスの ID・パスワードを管理するため, マスターパスワードの保護が重要になる. しかし, キーロガーによるマスターパスワードの不正窃取の危険性がある. キーロガーとは, キーボードの入力信号を記録し攻撃者のサーバ

に送信するものである．通信路を暗号化してパスワードを送信した場合でも端末の入力情報からパスワードを窃取されてしまう危険性がある．

2.2　既存方式のパスワードマネージャ

Web サーバに ID・パスワードを保存する SaaS 型の Web アプリケーションとしてパスワードマネージャが提案されている [2]．既存方式の概略図を図 2.2 に示す．

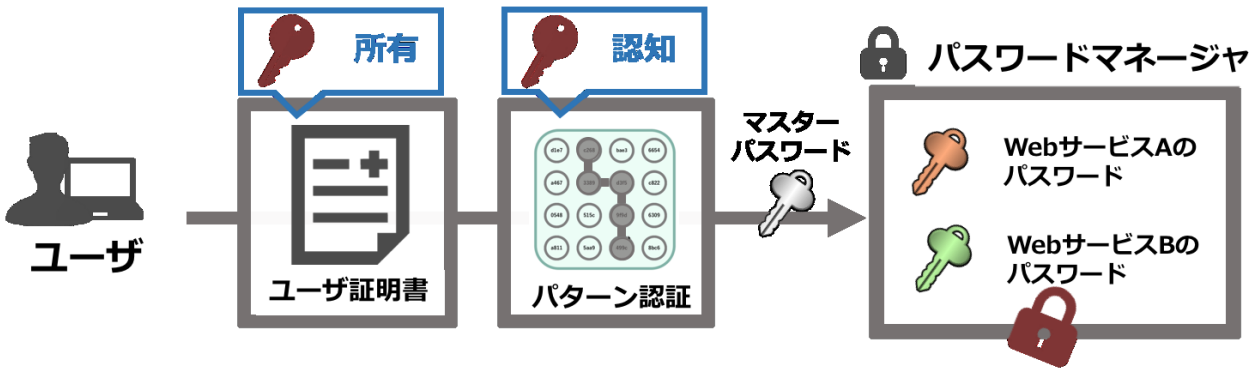

図 2.2　既存方式の概略図

既存方式はパターン認証の「認知」とユーザ証明書の「所持」の 2 つの要素を用いてマスターパスワードとする．ユーザ証明書内の情報からパターン認証の内部文字列を生成する．よってユーザが証明書を盗難したとしても正しいパターンを認知していなければ認証は行えない．逆に画面キャプチャ等によりパターンを窃取されてもユーザ証明書を所持していなければ認証は行えない．

2.3　パターン認証

パターン認証とは，マスとマスをドラッグで繋ぎ合わせることでパターンを生成し，その順番の正誤で認証を行う方式である [2]．ユーザが記憶するのは複雑な文字列ではなく，ユーザ自身が任意で設定したパターンであるため，利便性に優れる．また，パスワードは 12 桁以上の文字列がセキュリティレベルで推奨されており，4 × 4 のマスで安全性を満たす [3]．

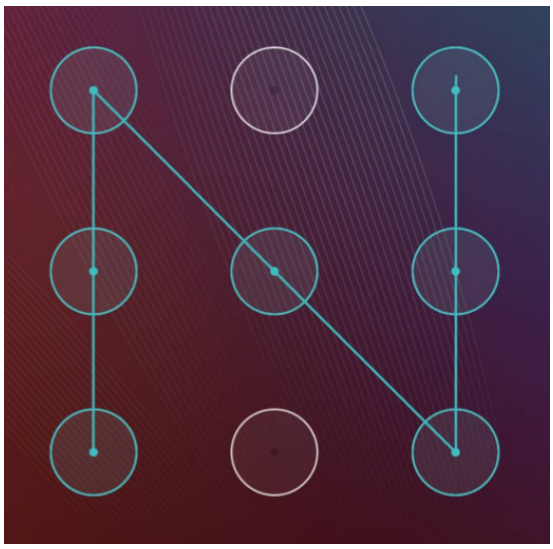

図 2.3　パターン認証の例

2.4　ユーザ証明書

ユーザ証明書とは, 正規の手順で登録されたユーザに対してサーバが発行するファイルである [2]. ユーザ証明書を所持していないユーザからのアクセスを防ぐことができるだけでなく, パスワードを記憶することなくファイルを送信するだけで認証できるメリットがある. 既存方式では, ユーザ証明書内にユーザ ID と乱数を保持する. 攻撃者が画面キャプチャ等により窃取したパターンを描写しても, ユーザ毎に証明書の内部文字列が異なるため, 他のユーザと二重で認証することはない [2].

2.5　認証情報

既存方式では, パターンのマス目の内部に文字列を格納し, それを連結した値を認証情報としている. 文字列はユーザ証明書に含まれる乱数から生成する. 生成方法は以下の通りである [2].

1. ユーザはユーザ証明書をサーバに送信する.
2. サーバはユーザ証明書から ID, 乱数を取り出す.

3. サーバが ID 毎に抛時している乱数のハッシュ値を適用する.
4. ハッシュ値を 4 × 4 マスに分散するめ 16 個に分解し, 格納する.
5. 格納したパターンを含むパターン認証をユーザに提供する

2.6　既存方式の問題点

既存方式の問題点は 2 つある. 1 つ目に認証時, 認証情報が毎回静的な認証情報が生成されていることである.　ユーザ証明書はユーザごとに異なり, 他のユーザと重複することはないが, ユーザ証明書とその内部文字列である乱数は毎回静的である. その乱数から認証情報を生成するため, 生成される文字列も静的である. 既存方式のパスワードマネージャは SaaS 型として提案されている. Web 上のパスワードマネージャは高いセキュリティが求められるため, 静的な認証情報では安全性が懸念される. 2 つ目に既存方式では,　ユーザ証明書を保存している 1 つの端末でしかパスワードマネージャを利用することができない. 近年は 1 人で複数の端末を利用するユーザが増加傾向にある. 既存方式のように複数の端末でパスワードマネージャを利用できないというのは, 利便性の低下を招いてしまう. 以上のことから, 取り上げた既存方式の問題点は以下の通りである.

- 認証時, 毎回静的な認証情報が生成されている
- 複数の端末でパスワードマネージャを利用することができない

第 3 章

提案方式

本章では, 前章で提示した問題点を解決する.

既存方式の問題点

- 認証時, 毎回静的な認証情報が生成されている
- 複数の端末でパスワードマネージャを利用することができない

提案方式

- 認証情報をワンタイムに変更して認証する
- QR コードによって端末間で認証に必要な情報を安全に共有する

3.1 表現と記法

下記に, 本章で用いる記法を示す.

- Server： 認証, データの管理, その他動的な処理を行うサーバ.
- User： サーバににリクエストを送るエンドユーザ.
- ID：ユーザ識別子.
- S：ユーザ, サーバで共有する秘密情報.
- N：サーバが ID を受け取ると生成する乱数. 認証の度に新しく生成する.
- P：ユーザがパターンをなぞることで生成した認証情報.
- H, F：一方向性関数を示す. 出力ビット数は常に一定.

3.2　認証情報の生成

提案方式で用いるパターン認証は既存方式と同じく, マスとマスをドラッグで繋ぎ合わせることでパターンを生成し, その順番の正誤で認証を行う方式である. 4 × 4 のマスで安全性を満たす [3]. 提案方式ではパターンのマス目に内部文字列として格納し, それを連結した値を認証情報とする. 文字列は秘密情報 S とサーバが認証の度に生成する乱数 N から生成する. 生成手順は以下の通りである.

1. User は ID を Server に送信する.
2. ID を受け取ったサーバは N を生成し, ユーザに送り返す.
3. ユーザ側では S と N の排他的論理和のハッシュ値を算出し, その文字列を 16 分割する (図 3.1).
4. 4 × 4 のマスに文字列を格納する (図 3.2).
5. ユーザは記憶しているパターンを入力することで, P を生成する (図 3.2).

例）$H(S \oplus N) =$
b2e766545b9da4675aa951ec05483389d8118bc66aa91ae3c822499c9385c288
↓
b2e7/6654/5b9d/a467/5aa9/51ec/0548/3389/
d811/8bc6/6aa9/1ae3/c822/499c/9385/c288

図 3.1　内部文字列の生成

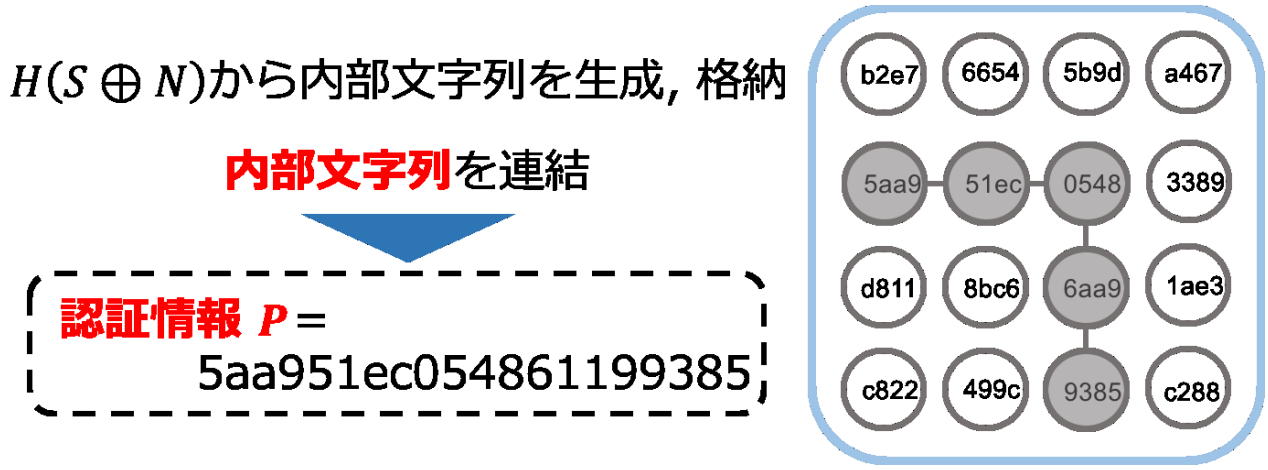

図 3.2　認証情報の生成

3.3　各フェーズ

提案方式を SaaS 型の Web アプリケーションとして提供する, User は Web ブラウザから提案方式にアクセスする.

3.3.1　初回登録フェーズ

User は提案方式を利用する前に, 初回登録を行う. 初回登録の流れを図 3.3 に示す.

1. User は ID と S を生成し, 末内に保存する. その後任意のパターンを入力する.
2. User は ID と S とパターンを Server に送信する.
3. Server は User から送られてきた ID と S とパターンを保存 User 登録を行う.

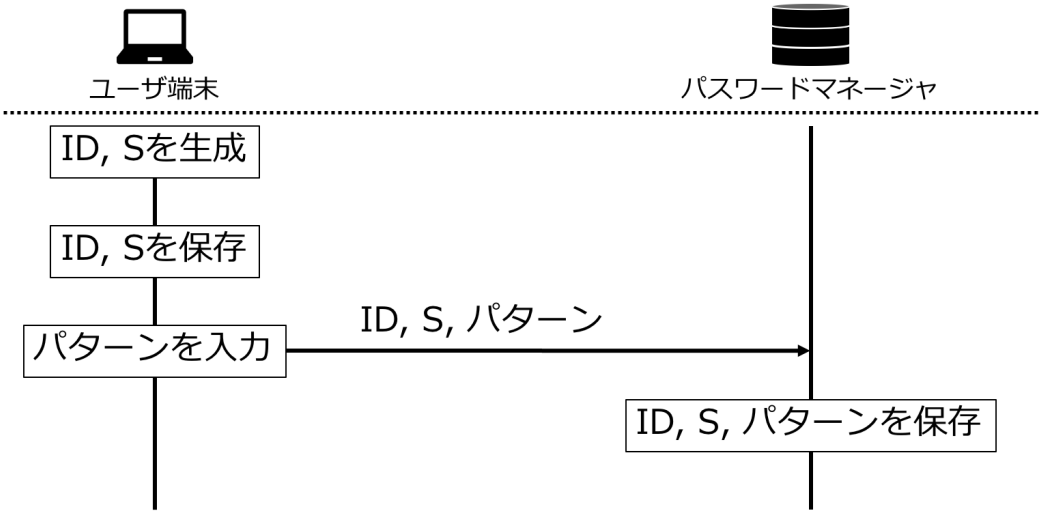

図 3.3　初回登録フェーズ

3.3.2　認証フェーズ

User が提案方式を利用する際に行う認証の流れを図 3.4 に示す.

1. User は認証リクエストとして ID を Server に送信する.
2. ID を受け取った Server は N を生成し, ユーザに送り返す.
3. User は端末内に保存している S と N からパターン認証のマスを生成する.
4. User は記憶する任意のパターンを入力することで P を生成し, H(P) を算出し, Server に保存する.
5. Server も S と N からパターン認証のマスを生成し, 保存しているパターンの通りに P を生成し, H(P) を算出する
6. Server は生成した H(P) と User から送られてきた H(P) の検証を行う.

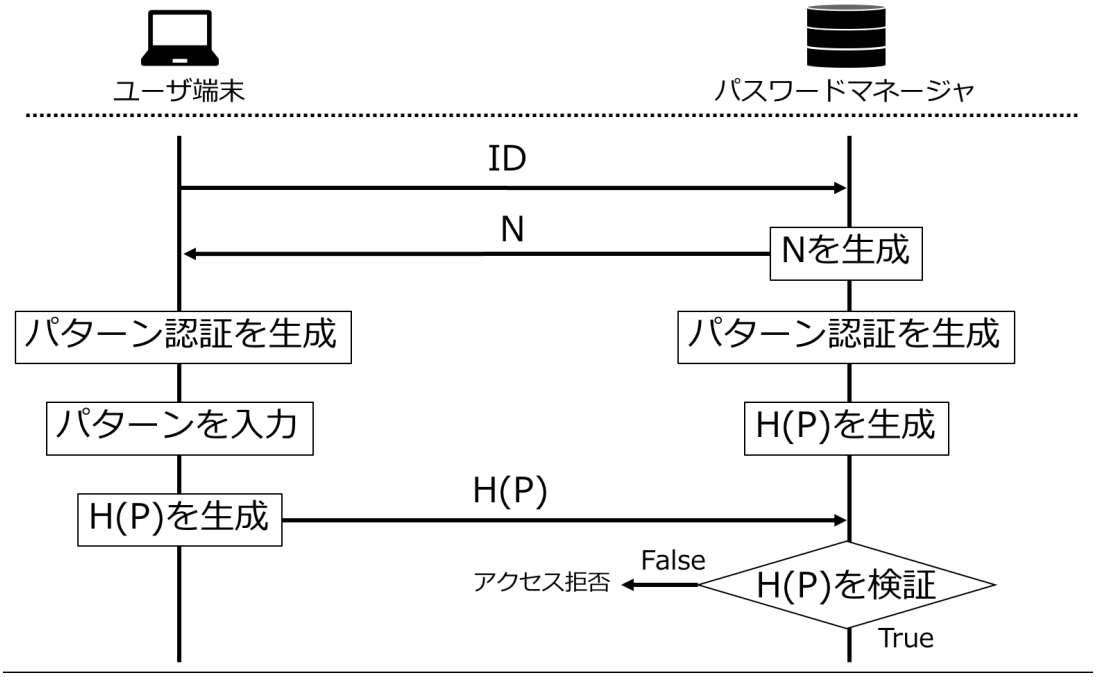

図 3.4 認証フェーズ

3.3.3 Web サービス登録フェーズ

Web サービスの ID・パスワードをパスワードマネージャに登録する流れを図 3.5 に示す.

1. User は認証フェーズ通りに認証を行う.
2. 認証後, User は登録したい Web サービス名とその ID・パスワードを入力する.
3. User は S と N の排他的論理和に 2 回ハッシュをかけた $F(H(S \oplus N))$ をセッションキーとして生成する.
4. セッションキーを用いて Web サービス名・ID・パスワードを暗号化し, Server に送信する.
5. Server もセッションキーを算出し, 複合を行い, Web サービスを登録する.

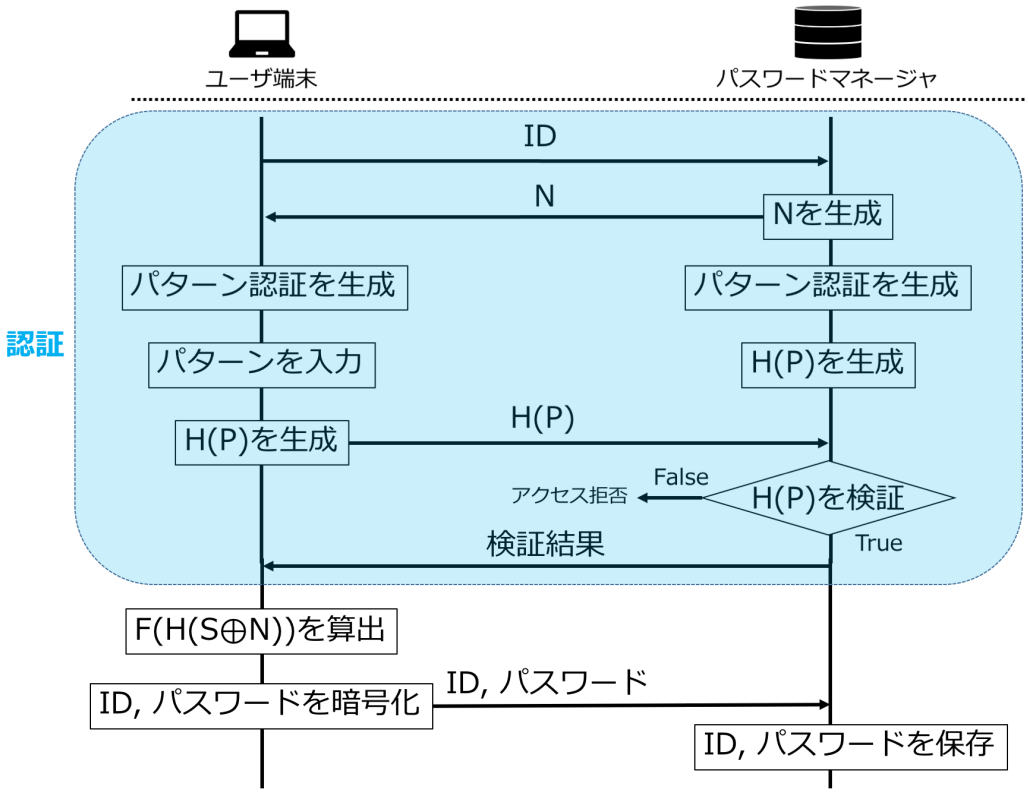

図 3.5　Web サービス登録フェーズ

3.3.4　利用フェーズ

認証フェーズ終了後，　パスワードマネージャに登録している Web サービスの ID・パスワードを取り出す流れを示す．

1. User は S と N の排他的論理和に 2 回ハッシュをかけた $F(H(S \oplus N))$ をセッションキーとして生成する．
2. User は利用したい Web サービス名を Server に送信する．その際，セッションキーで暗号化を行う．
3. Server は同じくセッションキーを生成，複合を行い，Web サービスに対応する ID・パスワードをセッションキーで暗号化し，User に送信する．
4. User はセッションキーで複合し，Web サービスをできるようになる．

3.3.5　複数端末化フェーズ

未登録の端末 B でパスワードマネージャを利用する流れを図 3.6 に示す．QR コードを用いることでインターネットを介することなく安全に情報を共有することができる [4].

1. すでに登録済みの端末 A の画面に, 端末内に保存している ID と S の 2 つから生成した QR コードを画面に表示する．
2. 端末 B の QR コードリーダを用いて QR コードを読み取り, 端末内に保存する．
3. 上記の認証フェーズと同じ認証を行うことでパスワードマネージャを利用する．

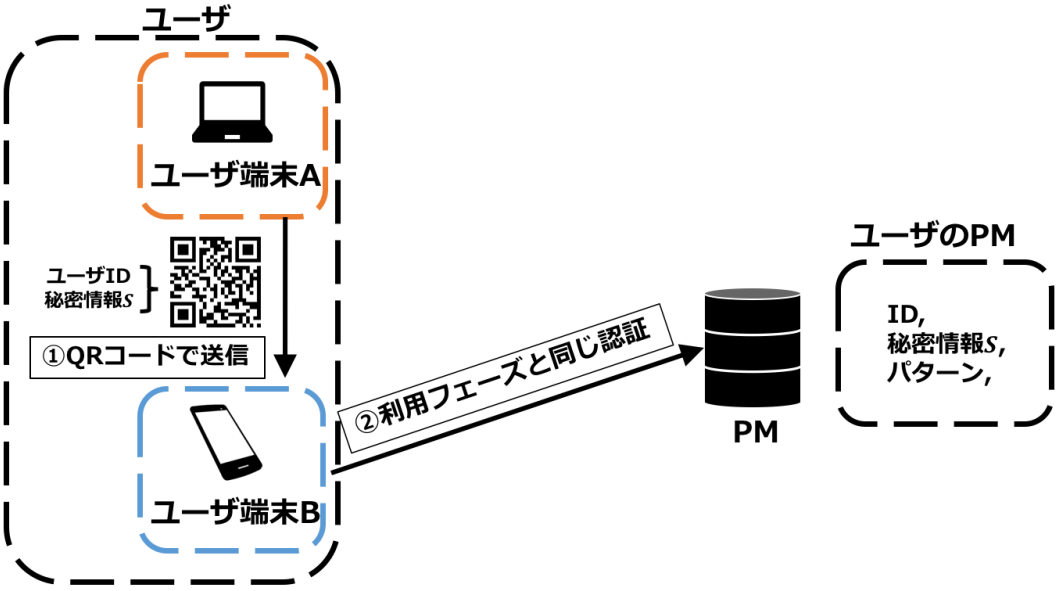

図 3.6　複数端末化フェーズ

第 4 章

評価

本章では, 既存方式と提案方式を比較し, 安全性と利便性の観点で評価を行う.

4.1 比較評価

表 4.1 は, 提案方式がパスワードマネージャへの攻撃に対して既存方式と同等の耐性を備えていることを示している. キー入力窃取に対しては, ユーザが入力するのは任意のパターンであるため, キー入力窃取の危険はない. 画面キャプチャに対しては, ユーザがなぞるパターンは盗まれてしまう可能性はあるが, 端末内の秘密情報が無いと認証情報を生成できないため, 問題は無いと考えられる. よって提案方式と既存方式共に

表 4.1　性能比較表 1

認証方式	キー入力窃取	画面キャプチャ
既存方式	不可	パターン
提案方式	不可	パターン

表 4.2 は, 提案方式と既存方式の安全性と利便性の比較を行う. 提案方式はパスワードマネージャへの攻撃に対して既存方式と同等の耐性を備えていることを示し, 認証情報生成部分の窃取による不正ログインの可能性は低いと言えるが, 提案方式では認証毎に異なる認証情報を生成しているため, 提案方式の方が安全性を向上することができた. また, 既存方式では複数の端末でパスワードマネージャを利用することができなかったが, 提案方式では複数の端末で利用できるようになり, 利便性を向上することができた.

表 4.2　性能比較表 2

認証方式	生成する認証情報	複数端末利用
既存方式	毎回同じ	不可
提案方式	毎回異なる	可

第 5 章

最後に

本研究は, 既存方式のパスワードマネージャの安全性と利便性を向上することを目的に提案を行った. 既存方式では, ユーザ証明書とパターン認証を用いる方式を提案していたが, 認証時に生成される認証情報が静的なデータになっていたため, 安全性に懸念があった. また, Web アプリケーションとして提案されていたが, 複数の端末でパスワードマネージャを利用することができす, 利便性の低下を招く問題があった. 提案方式では, 生成する認証情報を認証毎にワンタイムに異なるデータにすることによって安全性を向上した. 加えて複数端末の利用を可能とすることで利便性も向上した. よって既存方式よりも有用だといえる.

今後の課題として, 端末の紛失した場合の登録情報を結びつけたまま秘密情報の再発行を行う方法が必要だと考えられる. すでに複数の端末で秘密情報を共有している場合は良いが, 1 つの端末でのみパスワードマネージャを利用していた場合, その端末を紛失してしまうとパスワードマネージャが利用できなくなる問題が発生する. そのため登録情報を結びつけたままの秘密情報の再発行を行える方法が必要である.

謝辞

本研究の遂行及び論文の執筆にあたり，多くの御指導を頂きました高知工科大学情報学群清水明宏教授に御礼申し上げます．また，副査を担当して頂いた高知工科大学情報学群吉田真一准教授，植田和憲講師に御礼申し上げます．最後に，多くの御助言を頂きました高知工科大学情報学群セキュリティシステム研究室関係者各位に感謝申し上げます．

参考文献

[1] ICT 総研, 2017 年度 SNS 利用動向に関する調査, http://ictr.co.jp/report/20171011.html, 2018/2/25 閲覧

[2] 三本 拓也, ' パターン認証及びユーザ証明書を用いたパスワードマネージャの研究', 2017/2/28

[3] 石黒司，福島和英，清本晋作，三宅優, ' モバイル端末のロック解除向けパターン認証の安全性評価', 電子情報通信学会技術研究報告，Vol.112，pp.273-278，2012

[4] 株式会社デンソーウェーブ, コードドットコム QR コードとは?, http://www.qrcode.com/about/ 2018/2/25 閲覧